Ensemble Methods as a Defense to Adversarial Attacks against ML Models

Shimeng Chen* (20818524), Rongzhi Gu* (20855042), Chuxi Zhang (20815924)

ABSTRACT

Deep learning has been applied in many machine learning problems such as classification. A lot of counterexamples have shown deep learning is highly vulnerable to adversarial perturbations. For example, the neural networks applied on the self-driving car can be disturbed by small adversarial perturbations and misclassify some traffic signs. Hence, it is important to find some defence method to improve the robustness of deep learning systems. So, we explore the defence effect of some ensemble methods. And we test our method under the fast gradient sign method attack and AdvGAN. As for the fast gradient sign method attack, our methods can improve precision from around 30% to 80%. As for the AdvGAN, our methods do not have significant improvement.

1. INTRODUCTION

1.1 Problem Solved

Deep learning systems are sensitive to small adversarial perturbations. If the input image contains small adversarial perturbations on purpose, deep learning systems will give wrong prediction and cause serious accidents.

The aim of our project is to explore whether our defense strategies can make deep learning systems more robust when the system is attacked by some adversarial example. Our strategies include several different ensemble methods. And we will use fast gradient sign method (FGSM) and AdvGAN attack methods to test our defense strategies' effect.

1.2 Importance and Meaning

The trillion-fold increase in computing power has made deep learning (DL) widely used in processing various machine learning (ML) tasks, such as image classification, natural language processing, and game theory which makes DL occupied an important position in the field of computer vision, hence, ensuring the security and robustness of algorithms is of paramount importance [1].

However, researchers found that existing DL algorithms have serious security risks: attackers can easily deceive DL models by adding specific noise to benign samples, and they are usually undetected. Attackers use disturbances that are not perceivable by human vision/hearing, which is enough to make the normally trained model output high-confidence false predictions [2]. Researchers call this phenomenon an adversarial attack, which is considered to be a huge obstacle before deploying DL models in production. Therefore, research on adversarial attacks and defense technologies has attracted more and more attention from researchers in the field of machine learning and security.

* Authors contributed equally.

In the field of traffic safety, terrorists may be simpler and ruder, changing the prohibition of turning right to turning right, causing a traffic accident with only a flash. The essential reason is that the existing DL still fails to fully reproduce the process of human seeing things and cannot completely distinguish which features are useful for classification and which can be ignored [3]. It is conceivable that if this problem is not solved, the future application of DL to fields that require high reliability will inevitably bring a lot of security risks.

1.3 Why Other Methods May Not Work for This Problem

One simple method is called “adversarial training”, which means we should create and then incorporate adversarial examples into the training process. But this method is susceptible to a new class of attacks, the “blind-spot attack”, where the input images reside in “blind-spots” (low density regions) of the empirical distribution of training data but is still on the ground-truth data manifold.

Another defense method is to apply denoising autoencoders to preprocess the data used by the DNN. But denoising autoencoders are also vulnerable to adversarial attacks.

1.4 Why It is Better than Previous Methods

Ensemble methods mean constructing multiple classifiers used to classify new data points by the weighted or unweighted average of their predictions.

In this project, we try different data resampling ensembles such as Bagging, random split and K-fold Cross-Validation. In addition, we also explore the effect of combining several neural networks with different structures and the effect of adding gaussian noise into training dataset.

The advantage of our ensemble method is the diversity of classifiers. Since different classifiers have different decision boundaries, they perform quite differently on input images with adversarial perturbations. As for adversarial perturbations with small magnitude, the vast majority of classifiers were accurate. In that case, ensemble methods can improve the robustness of DNN.

This paper is organized as follows. In section II, we introduce two adversarial attack methods, Fast Gradient Sign Method and AdvGAN Method. In section III, ensemble method of defense adversarial attack is introduced. In section IV, we implement the ensemble defense system and reveal the experiment result. In section V, the conclusion is drawn. In addition, further work could be considered.

2. ADVERSARIAL ATTACK METHODS

2.1 Fast Gradient Sign Method

In our project, we use the fast gradient sign method (FGSM) introduced by Goodfellow et al. [4] as the adversarial attack method. The reason is that FGSM is a linear method to add perturbation thus provides a simple, cheap and fast way to generate satisfiable adversarial

examples. Moreover, FGSM is believed to have a considerable generalization across different neural networks and vast range of different training sets [4].

The added perturbation can be computed as:

$$\eta = \epsilon \operatorname{sign}\left[\nabla_x J(\theta, x, y)\right],$$

where

- θ: The parameters of a model.
- x: The input to the model.
- y: The targets associated with x (the original input label in our dataset).
- ϵ: The multiplier to ensure the perturbations are small.
- J: The lost function used to train the corresponding neural network.

Here in our project, the lost function is defined as Mean Square Error (MSE) and the gradient required in the formula can be efficiently computed with backpropagation.

The adversarial example thus can be generated by adding the above perturbations of small size $\epsilon > 0$ to the original input x. In our project, we try $\epsilon = 0, 0.01, 0.1, 0.15$ in experiments,

$$Adv_x = x + \epsilon\, sign\left[\nabla_x J(\theta, x, y)\right]$$

This method is simple and fast because it essentially finds how much each pixel in the original input image contributes to the current value of the lost function and adds a targeted perturbation according to its gradient computed using chain rule.

2.2 AdvGAN

Another way of generating adversarial examples to fool deep neural networks to produce wrong prediction is using generative adversarial networks (GAN) [5]. It can generate perturbations to the DNN models efficiently. It's believed that the adversarial examples generated by AdvGAN have better attack performance than other methods under regular defenses.

The overall architecture of AdvGAN can be described by Figure 1. It consists of three main compositions[5]. The generator $\mathcal{G}(x)$ takes as input the original instance x and generates its corresponding perturbation $\mathcal{G}(x)$ which is then added to x and sent to discriminator $\mathcal{D}$. The discriminator is used to distinguish the generated data and x. The basic principle of AdvGAN is generating examples which are indistinguishable with the data from its original class by the discriminator while are capable of misleading the target neural network f into predicting wrong labels.

In this report, the detailed network architectures of the discriminator, generator and target are omitted due to page limit. Further information is shown in our google colab codes.

3. Ensemble Methods

Ensemble method is a widely used technique to produce accurate and stable predictions. It combines several base machine learning algorithms or models to achieve better predictive performance than could be obtained by any of its constituents alone [6].

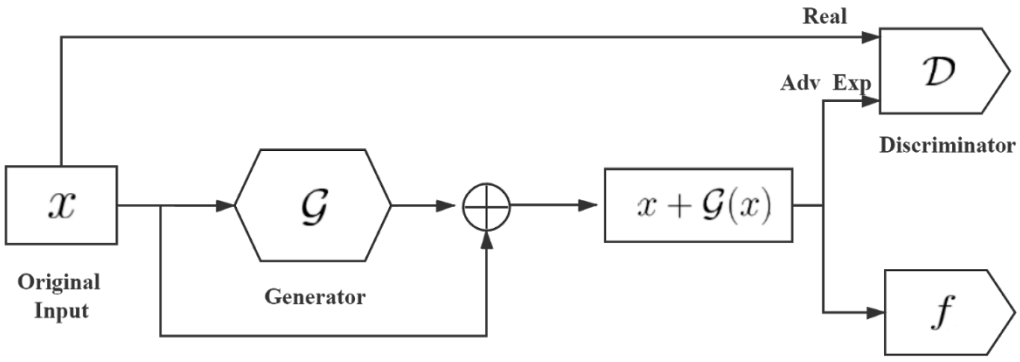

Fig. 1. Overall architecture of AdvGAN

In section 4, we use the following ensemble methods to test their defensive performance against adversarial attacks [7]. Effective and robust ensembles require each constituent to perform well individually but ideally disagree to some extent with each other. Therefore, these methods are similar but different in network layer or training datasets to encourage differences between ensembles. Note that ensemble model (c-f) are performed based on the Structure 1 introduced in model (a). The difference is that we encourage diversity between each classifier by using the same learning algorithm and structure on different training sets. Each classifier is trained using a resampled training data in turn to fit to different input bias and errors in order to achieve better stability and predictive performance.

a) Train multiple classifiers with the same network architecture but with random initial weights. Different initial conditions will lead to diverse final weights of the same classifiers which provide better diversity to the ensemble model. In our project, we use the architecture in Table 1, namely Structure 1 to train this ensemble method.

b) Train multiple classifiers with different neural network architectures. We modify the Structure 1 architecture by adding and removing some layers along with their parameters to introduce diversity to the classifiers. The architectures of the rest two classifiers are in Table 2 and 3, namely Structure 2 and Structure 3.

c) Inspired by RPENN [8], random split is used on the training data with which we train every classifier with a random sample of train dataset generated by train_test_split () function.

d) K-fold Cross-Validation is used on the training data with which the dataset is split into k sized folds equally and each fold has the opportunity to be the hold out and the model is trained on the remaining group.

e) Bagging [9] is used on the training data which draws m samples from the standard training set of size n uniformly and with replacement. A different bootstrap replicate is used as training data for each ensemble constituent.

Table. 1. Structure 1 Network Architecture

Layer Type	Parameters (59,210)
Relu Convolutional	32 filters (3×3)
Relu Convolutional	64 filters (3×3)
Relu Convolutional	64 filters (3×3)
Max Pooling	2×2
Dropout	0.2
Flatten	
Fully Connected	32 units
Dropout	0.2
Fully Connected	32 units
Dropout	0.2
Softmax	10 units

Table. 2. Structure 2 Network Architecture

Layer Type	Parameters (101,770)
Flatten	
Relu Fully Connected	128 units
Softmax	10 units

f) Add some small Gaussian noise to the training data so that all classifiers are trained on a similar but different training set. This method can somehow prevent the overfitting in each classifier and makes them more robust against adversarial perturbations.

In our project, we train three classifiers with 10 epochs using each ensemble method on their corresponding architectures and training datasets. The prediction of each ensemble is identified by letting each classifier vote for a label and taking the majority of votes as the output label.

4. Experiment

Table. 3. Structure 3 Network Architecture

Layer Type	Parameters (542,230)
Relu Convolutional	32 filters (3×3)
Max Pooling	2×2
Flatten	
Relu Fully Connected	100 units
Softmax	10 units

4.1 Experimental Setup

Platform: TensorFlow 2.3.0 provided by the google colab
Code link:
https://colab.research.google.com/drive/1qb_25Xpfbi1En9BRxrX199hX18m3WFjR?usp=sharing
Benchmark: The accuracy of prediction on test dataset from MNIST without adversarial attack and with adversarial attack

4.2 Dataset

In this project, our experiments are evaluated on the MNIST dataset [10]. The raw images from MNIST dataset are shown in Figure 2.

4.3 Results

Adversarial samples generated by FGSM introduced in section 2.1 corresponding to MNIST dataset can be found in Figure 3. All FGSM perturbations are generated with $\epsilon = 0.1$.

After we get these FGSM attack samples, we use them to test our networks trained by the original MNIST datasets, the adversarial performance of FGSM against the proposed network architectures is shown in Figure 4.

For simple and naive approach, we include the attack samples into our training dataset and then train a new model with a new training dataset. The base precision is 97.75% while the precision under attack is 0.23%, as shown in the Figure 5. So, we think the naive approach is useless.

The performance of ensemble defense approach against FGSM is shown in the Figure 6 where the sky-blue bar represents the precisions of our model under the FGSM perturbation generated by the gradient of one of DNN. The green bar represents the FGSM perturbation generated by the average of the gradient of three DNN. We can see that as for FGSM attack, m2 and m7 is more robust than other methods.

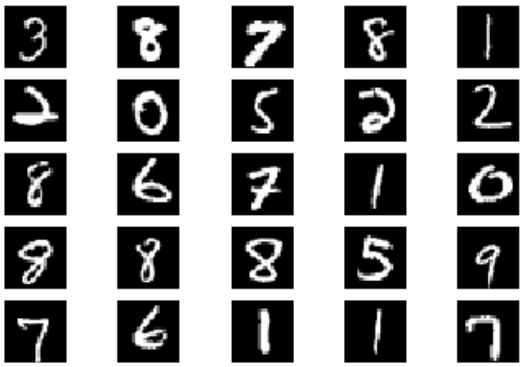

Fig. 2. Raw images from MNIST dataset

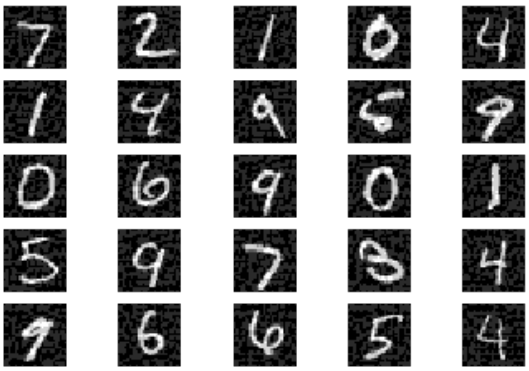

Fig. 3. FGSM attack sample image

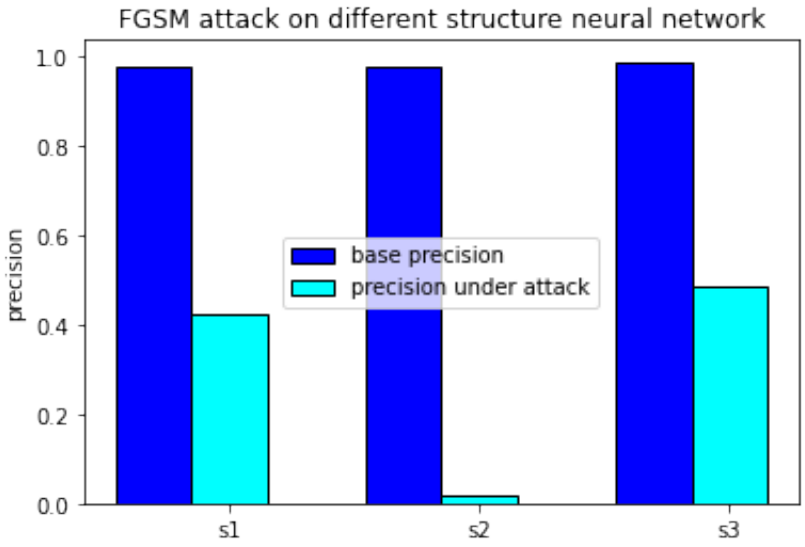

Fig. 4. FGSM attack on different structures

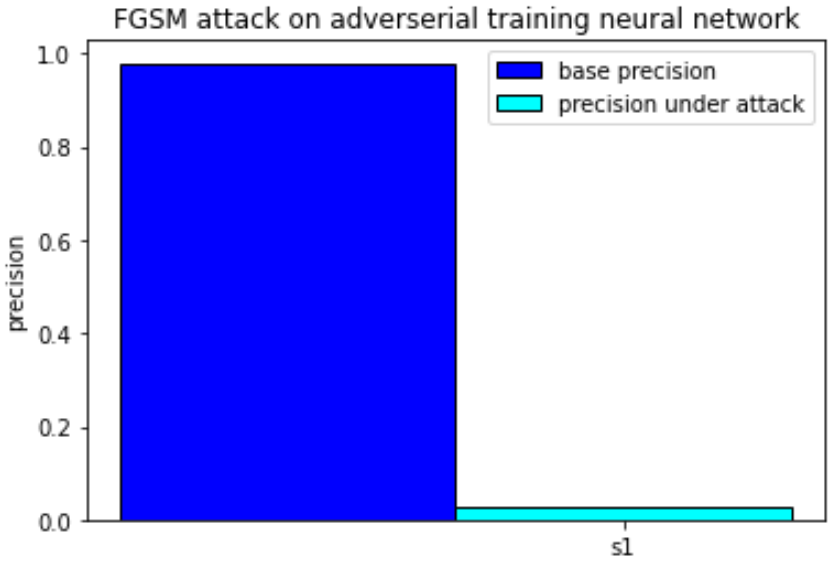

Fig. 5. Naive defense approach

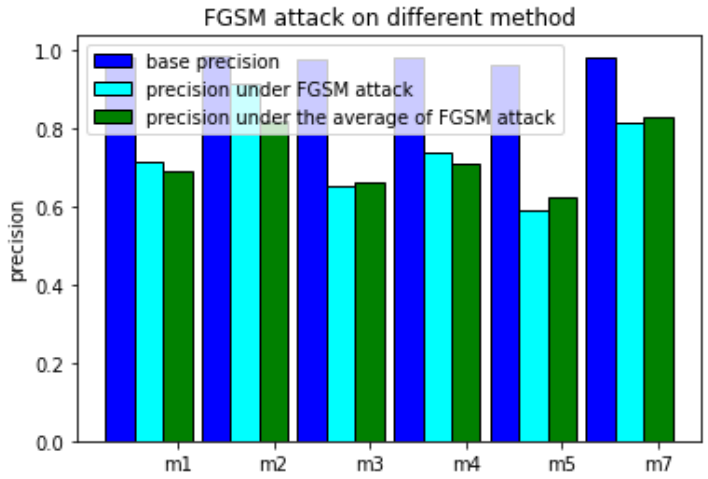

Fig. 6. Ensemble defense approach against FGSM. **m1**: Train multiple classifiers with the same network architecture but with random initial weights. **m2**: Train multiple classifiers with different structure neural network architectures. **m3**: Random split ensemble which trains every classifier with a random sample of train dataset generated by train_test_split () function. **m4**: K-fold Cross-Validation Ensemble. **m5**: Bagging ensemble. **m7**: Add some small Gaussian noise to the training data so that all classifiers are trained on a similar but different training set.

The perturbated images generated by AdvGAN can be shown in Figure 7. It can be noticed that the perturbations added are more noticeable than that of FGSM.

Before we applied the perturbated images to our ensembles, the adversarial performance of advGAN against the proposed network architectures trained by original datasets is shown in Figure 8. Then, as shown in Figure 9, our ensemble methods do not have a good defense performance against those adversarial examples generated by advGAN.

5. CONCLUSION AND FUTURE WORKS

5.1 CONCLUSION

As for FGSM attack, the defense effect of our ensemble methods works well. Because different models have different decision boundary and their gradients are different. As for Adv-GAN attack, the precision of our model decreases a lot from above 90% to around 30%. Since Adv-GAN perturbation is more noticeable than that of FGSM, its perturbation may have larger magnitude, which may be the reason why it is hard to defend.

5.2 FUTURE WORKS

Due to the limited time, we haven't found a way to transfer our models from TensorFlow to other platforms. If there is more time available, we may use symbolic methods [11] to test our defense method and find its weakness. In addition, we may use SMT based methods to analyze weaknesses of our methods, such as Reluplex and Marabou framework.

Since the decision boundary of modern deep neural networks are piecewise linear, they may be attacked easily by some perturbation. If we use some quadratic model such as shallow

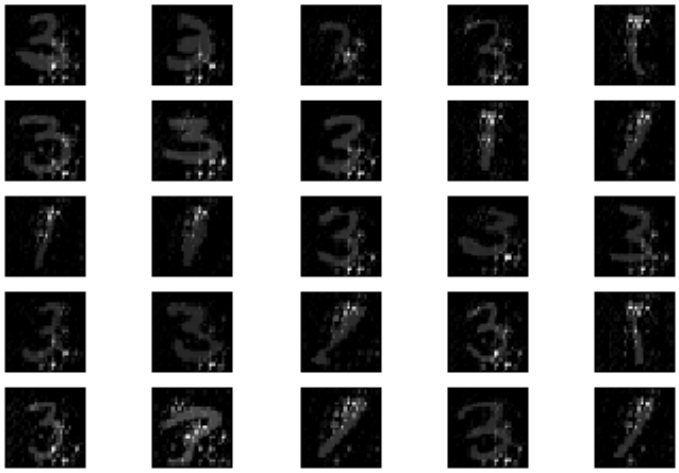

Fig. 7. AdvGAN attack sample image

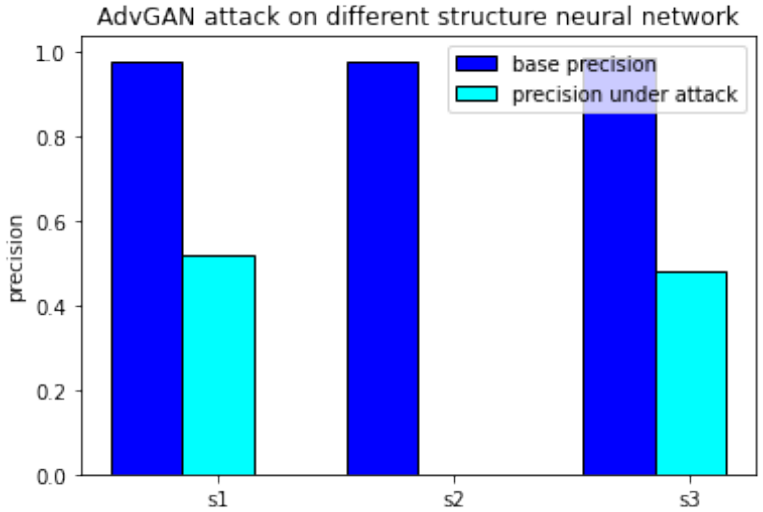

Fig. 8. AdvGAN attack on different structures

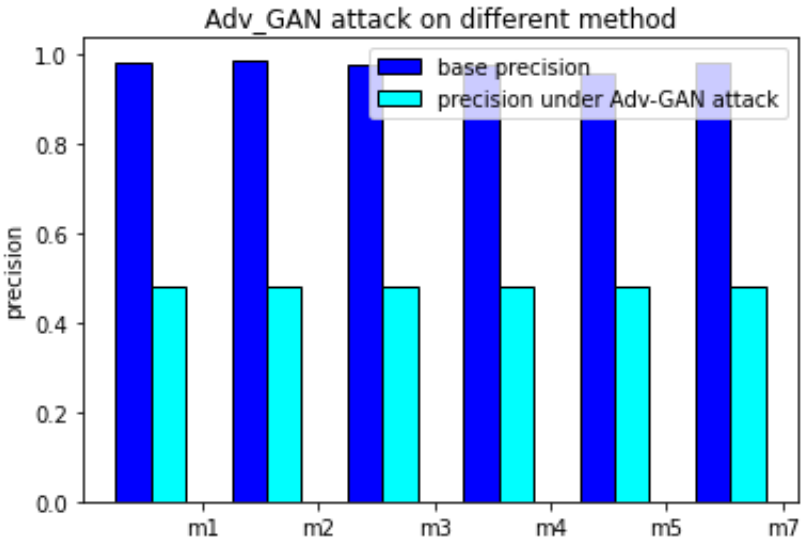

Fig. 9. Ensemble defense approach against AdvGAN

RBF models, we may make our ensemble model more robust. The weakness of shallow RBF models is their poor accuracy. If we combine shallow RBF models with deep neural networks by ensemble learning, we may get a new model with good precision and robustness.

6. REFERENCES

[1] Yuan, X, He, P, Zhu, Q. and Li, X, Adversarial Examples: Attacks and Defenses for Deep Learning, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805-2824, 2019.

[2] Zhu, Dingyuan, Zhang, Ziwei, Cui, Peng, and Zhu, Wenwu, Robust Graph Convolutional Networks Against Adversarial Attacks. *In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[3] Hidano, Seira, Takao Murakami, Shuichi Katsumata, Shinsaku Kiyomoto, and Goichiro Hanaoka. “Model Inversion Attacks for Prediction Systems: Without Knowledge of Non-Sensitive Attributes.” 115–11509. *IEEE*, 2017.

[4] Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[5] Xiao, Chaowei, et al. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

[6] Rokach, L, Ensemble-based classifiers. *Artif Intell Rev* 33, 1–39, 2010.

[7] Strauss, Thilo, Hanselmann, Markus, Junginger, Andrej, Ulmer, Holger. Ensemble Methods as a Defense to Adversarial Perturbations Against Deep Neural Networks, *arXiv:1709.03423*, 2017.

[8] Potdevin, Yannik, Nowotka, Dirk, Ganesh, Vijay. An Empirical Investigation of Randomized Defenses against Adversarial Attacks, *arXiv:1909.05580*, 2019.

[9] Breiman, Leo. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[10] LeCun, Yann, Bottou, Leon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] Gopinath, Divya, Wang, Kaiyuan, Zhang, Mengshi, Pasareanu, Corina S, and Khurshid, Sarfraz. Symbolic execution for deep neural networks. *arXiv preprint arXiv:1807.10439*, 2018.